

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**

## マルチギガビット IP フォワーダ用

### 50Mpps Longest Prefix Match 検索 LSI

小林正好<sup>†</sup> 村瀬勉<sup>†</sup> 小倉直志<sup>‡</sup> 栗山敦<sup>‡</sup>

<sup>†</sup>NEC C&C メディア研究所

<sup>‡</sup>NEC LSI 事業本部システムマイクロ事業部

<sup>†</sup>〒216-8555 川崎市宮前区宮崎 4 丁目 1-1

<sup>‡</sup>〒211-8666 川崎市中原区下沼部 1753

E-mail: masayoshi@ccm.CLnec.co.jp, murase@ccm.CLnec.co.jp,

ogura@LSInec.co.jp, akuri@LSInec.co.jp

あらまし 本稿では、OC-48 等のマルチギガビットリンクのパケットを、QoS 制御をしながら、ワイヤスピードでフォワード出来る IP フォワーダの構成について述べ、フォーディングテーブル検索に必要なとされる検索エンジンへの要求条件を述べる。そして、検索データ幅が広く、常に一定時間で高速に Longest Prefix Match 検索が可能で、かつ、ルーティングテーブル更新にも容易に対応できて、大きなエントリ数を扱える、といった要求条件を満たすことが可能な、CAM(Content-Addressable Memory)ベースの Longest Prefix Match 検索 LSI のアーキテクチャを提案する。このアーキテクチャによって最大 50Mpps の Longest Prefix Match 検索を実現可能である。本稿の最後には、このアーキテクチャを実装し、実際に開発した LSI の仕様を述べる。

キーワード Longest Prefix Match, IP ルーティングテーブル, テーブル検索, 高速ルータ

## A 50Mpps Longest Prefix Match Search Engine LSI for Multi-gigabit IP Forwarding

Masayoshi Kobayashi<sup>†</sup> Tutomu Murase<sup>†</sup> Naoyuki Ogura<sup>‡</sup> Atsushi Kuriyama<sup>‡</sup>

<sup>†</sup>C&C Media Research Laboratories, NEC Corporation

<sup>‡</sup>Network LSI Department, System Micro Division, NEC Corporation

<sup>†</sup>1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa 216-8555, Japan

<sup>‡</sup>1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, Japan

E-mail: masayoshi@ccm.CLnec.co.jp, murase@ccm.CLnec.co.jp,

ogura@LSInec.co.jp, akuri@LSInec.co.jp

**Abstract** In this paper, we present several requirements for search engines to forward IP packets with QoS controlling at a wire-speed of a gigabit-link such as OC-48. And we propose a CAM(Content-Addressable Memory) based LSI architecture which can perform Longest Prefix Match search at a speed of 50 million cycles per second. This architecture meets all of the requirements, such as, (1) wide data width, (2) taking small fixed clock ticks regardless of keys to search, (3) easy to follow routing table changes and (4) accommodating with large number of entries. It is possible to achieve 50 Mpps (packets per second) of Longest Prefix Match search with the proposed architecture. We also present the manufactured LSI which implements our searching method.

**Keywords** Longest Prefix Match, IP routing table, table search, fast router

## 1. はじめに

近年の IP トラフィックの急激な増大に追従するため、ネットワークの物理リンク速度は向上し、1Gbps 以上のリンク速度の、ギガビットイーサネット(IEEE 802.3z), OC-48(2.4Gbps)などが登場している。さらに、近い将来 OC-192(9.6Gbps)レベルのインタフェースの開発も積極的に議論されており、このようなリンク速度に追従できるマルチギガビット IP フォワーダの開発が急務となっている。

IP パケットフォワーディング処理では、宛先 IP アドレスをもとに Next ホップを決めるためのフォワーディングテーブル検索が最大のボトルネックである。これは、CIDR(Classless Inter-Domain Routing)を行うため、IP フォワーディングテーブルの検索に、複雑な Longest Prefix Match 検索が必要だからである。これまでも、Longest Prefix Match(以下 LPM)検索に関して、様々な高速化の手法が研究され[3,4]、スループットも、平均的にはマルチギガビットのリンク速度に近づいてはきたが、検索時間にばらつきが大きく、常にリンク速度のスループットを達成するということが出来なかった。

本稿では、従来のこうした問題点を解決する、LPM 検索チップのアーキテクチャと、実際に LSI 化したチップの性能を示す。まず、2 章ではワイヤスPEEDでのマルチギガビット処理を可能にするハードウェアによるフォワーダの構成について述べ、その中で LPM 検索エンジン(サーチャ)に必要とされる機能、性能を説明する。3 章では、LPM 検索について、従来の検索方法の問題点を述べる。4 章ではサーチャへの性能要求条件をまとめ、5 章で 50Mpps(packets per second)の検索性能をもつサーチャの動作原理と実際に開発した LSI の性能を示す。

## 2. マルチギガビットフォワーダ

マルチギガビットリンクに対し、ハードウェアで、ワイヤスPEEDの IP フォワーディング処理を行うフォワーダの構成について述べる。ここで、ワイヤスPEEDとは、最小長さの packets がリンクの帯域 100% で到着するのと同じ頻度で、常に処理を完了する事を意味する<sup>1</sup>。

### 2.1. IP パケットのフォワーディング処理

フォワーディング処理の高速化のポイントを明らかにする為、まずルータのフォワーディング処理について説明し、ルータ処理のボトルネックを明確にする。

ルータはパケットを受信すると、パケット長、チェ

ックサム等进行检查する。正当な IP パケットと判断されると、宛先 IP アドレスを取り出し、後述する LPM 検索でフォワーディングテーブルの検索を行い、出力すべきインタフェース情報を得る。最後に、TTL の減算/チェックサム再計算の処理を行い、パケットを出力インタフェースへフォワードする。また、QoS 制御処理に対応するため、上位ヘッダも併せて検索する機能が求められる。

ハードウェアでフォワーダを構成する場合、これらの処理の中で、ヘッダの正当性検査など、テーブル検索処理以外の処理は必要な計算量が小さく、並列化が可能となす。高速化は容易である。しかし、テーブル検索処理は検索アルゴリズムが複雑で計算量が大い上、検索対象であるテーブルの容量が大きく、並列化が困難で、フォワーディング処理のボトルネックとなっている[7]。

### 2.2. パイプライン化 IP フォワーディング処理

フォワーディング処理の高速化には、まずボトルネックであるテーブル検索処理部分の利用率を高める必要がある。このため、フォワーディング処理を、検索処理部が独立するよう分割し、他の部分とパイプライン動作させることにする。以下では、パイプライン分割について説明し、サーチャへの要求条件について述べる。

最小の長さの packets をリンク速度で伝送するのに必要な時間を 1 パケット時間と呼ぶことにする(OC-48 リンクで約 150ns)。ワイヤスPEEDでのフォワーディングを行うためには、パイプラインの各ステージが、常に 1 パケット時間内で処理を終える必要がある。この「常に 1 パケット時間以内」という条件は重要で、これが満たされないと、ワイヤスPEEDフォワーディングを達成できない。

そこで、(1)検索処理部が他の部分と独立する、(2)各ステージが 1 パケット時間内で処理を終えることが出来る処理内容にする、という 2 点に留意して、例えば図 1

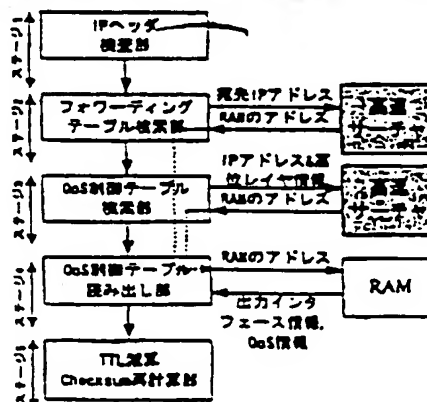


図 1: ハードウェアによるパイプライン化 IP 処理

<sup>1</sup> 本稿では、IP ルーティングプロトコルから得られるルーティングテーブルと、検索用のデータ構造を用いてそれらを格納したフォワーディング用のテーブル(フォワーディングテーブル)を区別する。

<sup>2</sup> IP-on-the-fly などでも知られる。カットスルー処理[2]を行う場合、バッファリングが出来ないため、ワイヤスPEEDでフォワード処理することは必須である。

<sup>3</sup> IP over SONET/SDH の OC-48 をリンク速度でフォワーディングするのに要求される検索時間は、最小パケット長を IP ヘッダ(20byte)+TCP ヘッダ(20byte)+PPP のカプセル化(8 bytes)=48 bytes として計算している(厳密には約 15432 ns)

に示すように5ステージのパイプラインに分割した構成をとることで処理全体の高速化が図れる。

検索処理部については、後述するように、従来の検索手法では検索速度が遅かったり、検索時間にばらつきが大きく、「常に1パケット時間」という制約を満たせない。このため、本稿では新たに、ギガビット級リンクに対し常に1パケット時間内で検索を終えることができる、高速なサーチを提案する。

### 2.3. QoS制御に必要なサーチのデータ幅

レイヤ4以上のヘッダ情報をもとにパケットのQoS制御を行う為に必要とされる、検索処理について説明し、高速サーチへの機能要求条件を追加する。

ルータにはフォワーディング機能のみならず、QoS制御機能のサポートが必須となりつつある。このため、TCP/UDP等のヘッダまで見て、IPアドレスとTCP/UDPポート番号の組や、IPヘッダのTOS(Type of Service)バイトに応じて出力優先度や出力レート異なるキューへ格納する機能が必要である。例えば、特定のサブネットからのWebトラフィックを優先するためには、TCPのプロトコル番号(8bit)とポート番号(16bit)を、サブネットを示すIPアドレスのプレフィックスと組にしたエントリをテーブルに登録しておき、検索する必要がある。この検索データはプレフィックスを含むため、LPM検索の必要があり、LPM検索エンジンはこの検索にも使用できるべきである。このためには、高位レイヤヘッダ情報をIPプレフィックスと組にして格納するために、検索データの幅が32bit(IPv4の場合)より広い事が要求条件となる。

## 3. IP フォワーディングテーブル検索処理

### 3.1. Longest Prefix Match 検索

IP フォワーディングテーブル検索処理の性能について議論するために、まずIP フォワーディングテーブル検索に特有の処理であるLPM検索について説明する。

IP フォワーディングテーブルは、表2のように、宛先フィールドとそれに対応する出力インタフェース情報フィールドの組から構成される。被検索対象となるのは宛先フィールドで、プレフィックスと呼ばれる、IPアドレスをMSBから任意の長さのビット分だけ指定したビット列で表現される。指定していないビットは'1'でも'0'でもよいという意味である。表1では32bit IPアドレスを8ビットずつドットで区切り、それぞれを10進数で表現している。プレフィックスで指定しないビットに対応する10進数部分は、'\*'で表現している。LPM検索では、テーブルの宛先フィールドとパケットの宛先IPアドレスを、エントリのプレフィックスで指定した部分にのみ着目して比較し、一致するエントリのうち(一致するエントリは複数あり得る)、最も長いプレフィックスを持つエントリが検索結果となる。例えば、パケットの宛先IPアドレスが、10.1.2.3である時、表2を検索すると、宛

先フィールドのプレフィックスが一致するものは、10.1.2.\*と、10.1.\*.\*の2エントリであるが、この中で最長のプレフィックスを持つのは10.1.2.\*であり、これが検索結果のエントリである。

表1: IP フォワーディングテーブルの例

元元	出力インタフェース情報	
	Interface 番号	次ホップ IP address
10.1.2.*	Interface #1	10.100.1.1
10.1.*.*	Interface #2	10.100.2.1
10.2.*.*	Interface #3	10.100.3.1
*.*.*.*	Interface #4	10.100.4.1

### 3.2 従来の検索方法 問題点

LPM 検索に対する高速化については様々な研究がなされており、これらは大きく分けて、

(A) Patricia Tree などの Tree 構造のデータ構造を用いた検索方法[8]、

(B) 高速なプロセッサとそのキャッシュメモリにフォワーディングテーブルを格納する方法[3,4]

(C) CAM を使う方法[5,6]

の3つがある。それぞれについて概要と問題点を述べる。

(A)は、フォワーディングテーブルを Tree 構造で表現し、Tree をたどって検索を行ってゆく方法である。この検索方法では、Tree をたどる時の枝分かれ毎に、比較判定と、時間のかかるメモリアクセスが必要のため、検索時間がかかり、マルチギガビットのリンクを扱えない。さらに、ルーティングテーブルが変化した場合、Tree の再構成に多くの時間がかかるという欠点がある。

(B)近年、高速な汎用プロセッサのキャッシュメモリにフォワーディングテーブルをコンパクトに格納する方法が研究されている[1,2]。これらの方法は、平均検索時間としては、マルチギガビットのリンクをフォワードするに足る性能を達成しつつあるが、検索のキーとして与えるIPアドレスと、フォワーディングテーブルの組み合わせによって、検索時間にばらつきが大きく、最悪検索時間ではリンク速度を達成するほどの性能は出ていない。また検索時間にばらつきがあることは、フォワード処理のパイプライン構成が非常に複雑になるという問題もある。さらに、(A)と同様、フォワーディングテーブルの再構成に時間がかかる欠点もある。

(C)ハードウェアで完全一致検索の出来るCAMを用いた方法も研究されている。この方法には、エントリをプレフィックス長毎に分けて異なるCAMに格納しておき、検索でヒットしたCAMの中で、最大長のプレフィックスをプライオリティエンコードで選り出す方法[5]や、CAMに付加回路をつける方法[6]などがある。しかし、いずれの場合も、マルチギガビットのリンクを扱えるほど高速ではない。

## 4. 要求条件

以上の議論をふまえ、マルチギガビットフォワード用の

כח 452

- フィクス部以外のビットには'0'でも'1'でもよいが、仮に'0'を埋めるものとする。マスクビット列はプレフィクス部を MSB 側から連続して'1'を格納して表現するものとする。たとえば、8bit の検索データ幅で、5MSB が10100 であるというプレフィクスを表現するなら、データビット列には 10100000、マスクビット列には 11111000 と格納される。このようにデータが格納された状態で、検索の1段階目、2段階目ではそれぞれ以下の様な動作が起こる。

## 5. 超高速 Longest Prefix Match 検索 LSI

- 1-1. 図2の検索データ F/F(Flip Flop)が検索データをラッチし、検索データが各エントリに伝わる。
- 1-2. 図2の各エントリでは、検索データとプレフィックスの比較を行う。すなわち、検索データとデータビット列を、マスクビット列の'1'が立っているビットにのみ着目して比較する。そして、比較結果に従い、表2の様に図2のプレフィクス一致線、マスクビット列出力線に値を出力する。
- 1-3. Shortest Mask 線では、各エントリのマスクビット出力線の、エントリ間に対応するビット同士で Wired OR が取られる。この結果、n 本の Shortest Mask 線には、プレフィクス比較結果が一致するエントリのうち、プレフィクス長(マスクビット列の中で MSB から'1'が連続しているビット数)が最大のマスクビット列を持つエントリのマスクビット列である、Shortest Mask が現れる。

- 2-1. 1 段階目で得られた Shortest Mask が各エントリに入力される。
- 2-2. 各エントリでは、1 段階目とは異なり、データビット列ではなく、マスクビット列と Shortest Mask の値を完全一致比較し、結果を図 2 の Longest Mask 一致線に出力する。
- 2-3. 2-2 の比較の結果、一致したエントリで、かつ、1

図2にLSIのブロック図を示す。図のように、各エントリはサーチの検索データ幅のデータビット列とマスクビット列をもっている。このデータビット列とマスクビット列の組は3.1で説明したプレフィクスを表現し、データビット列にはプレフィクスが、マスクビット列にはデータビット列のうちのプレフィクス部を示すビット列が格納される。ただし、データビット列のうち、プレ

段階目のプレフィクス比較結果が一致しているエントリは、Encoder に'1'を出力する。それ以外のエントリは'0'を出力する。この結果 Encoder には、LPM したエントリのみが Encoder に'1'を出力することになる。

- 2.4. Encoder から、検索結果(LPM したエントリのアドレス)が出力される。

表2: 各エントリの1段階目の出力

プレフィクス比較結果	プレフィックス一致線	マスクビット列出力線
不一致	0	ALL 0
一致	1	マスクビット列

各エントリには、1 段階目では検索データ、2 段階目では Shortest Mask が入力として与えられ、1-2、2-2において、それぞれ、図2のプレフィックス一致線、Shortest Mask 一致線の値を決定しているが、この部分の動作は一般の CAM の動作と同様で、回路構成も CAM と同様の回路構成が取れる。各エントリのマスクビット列について、エントリ間で Wired OR を取る部分が一般の CAM と大きく異なり、これによって従来の CAM を使った LPM 検索方式と比べ、検索速度を大幅に向上させている。

ここで、この LSI アーキテクチャの動作周波数について考察し、検索性能について述べる。提案する LSI のアーキテクチャでは、1 段階目最初の clock の立ち上がりから Shortest Mask が確定するまでの遅延  $T_1$  (図3参照)が大きく、これが動作周波数を決定する。この遅延値  $T_1$  は、1 段階目 1-2 のプレフィックス一致線が確定するまでの遅延  $t_1$  と 1-3 の Wired OR で Shortest Mask が確定するまでの遅延  $t_2$  の合計である。前者の遅延値は検索ビット幅に依存し、後者はエントリ数に依存する。また、これらの遅延値は LSI のテクノロジーにも依存する。4K エントリで、64bit データ幅の場合、現在の一般的 LSI 製造技術である、0.25  $\mu$  デザインルール拡散製造プロセスでの、回路シミュレーション結果の遅延値、 $t_1=2.5ns$ 、 $t_2=8.1ns$  を用いると  $T_1=2.5+8.1=10.6ns$  となる。2 段階目の遅延  $T_2$  は  $T_2 < T_1$  であることを考慮し、実際の遅延値は LSI 上の配線、レイアウト等に左右されることを考慮しても、 $T_1+T_2 < 40ns$  を満たすと考えられ、本アーキテクチャの LSI は動作周波数 25MHz (1clock=40ns) 程度では十分動作すると考えられる。

また、図2のA,Bにおいて F/F (flip-flop) を挿入すると、1 段階目、2 段階目で使用する構成要素を分けるとが出来、動作周波数を上げて検索をパイプライン実行できる

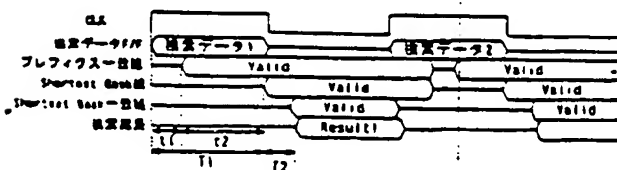


図3: タイミングチャート

ことから、F/F を挿入しない場合に比べ、検索のスループットを向上させる事ができる。F/F を挿入してパイプライン動作をした場合のタイミングチャートが図4で、2つの検索データを連続してパイプラインで検索している場合を示している。この場合、動作速度は先と同様の条件で考えると、 $T_1=10.6ns$  かつ  $T_2 < T_1$  であり、1clock=20ns の 50MHz で動作すると考えられる。

以上から、このアーキテクチャの検索性能は、F/F 挿入を行わない場合は、検索時間 40ns (25MHz, 1clock) で検索スループットは 25Mpps、F/F を挿入した場合は、検索時間 40ns (50MHz, 2clock) で、検索スループットはパイプライン検索で毎クロック検索できるので 50Mpps となる。

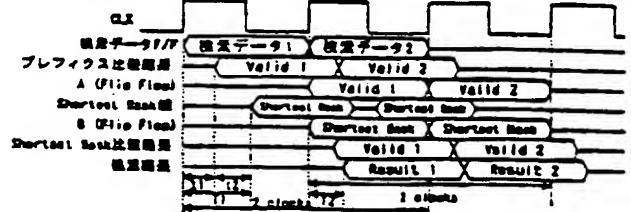


図4: F/F を挿入した場合のタイミングチャート

## 5.2 エントリ数の拡張

5.1 で示した検索 LSI を複数接続してエントリ数を増加することが出来る。以下では、5.1 で F/F を挿入した場合の LSI の接続について述べる。

LSI を複数接続するには、LSI 外部への出力である、図2の Shortest mask 出力同士 (図2のC点) をチップ間で Wired OR 接続すればよい。この Wired OR 接続によって、5.1 の1段階目 1-3 の Wired OR が複数チップに及ぶため、複数接続チップ間で、Shortest Mask を得ることが出来る。この共通の Shortest Mask が各チップの2段階目での検索対象となるので、複数チップの全エントリに対する LPM 検索が可能となる。

複数チップを接続する場合には Wired OR がチップ間に及ぶため、 $t_2$  (Wired OR の収束時間) の値が接続するチップの数に依存する。 $T_1=t_1+t_2$  の値が 1clock 未満であれば、5.1 と全く同じ動作になるが、1clock 以上かかる場合には、図2の、B の F/F の確定を遅らせて対応する。1 段階目の最初の clock 立ち上がりから Shortest Mask が収束するまでのかかる clock 数を  $D$  とすると、 $D=2$  の場合の Timing Chart は図5のようになる。この場合、検索時間は  $D+1$  clock かかり、検索データは  $D-1$  clock 開けて次々に検索することが出来る。

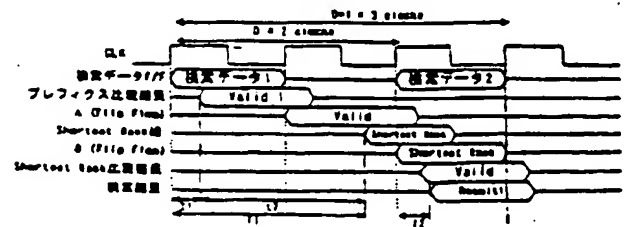


図5: 複数 LSI 接続の場合のタイミングチャート

D の値を決定する Wired OR の遅延時間は、Wired 接続を行う端子、配線の容量にほぼ比例する。ただし、後述するように軍用的なエントリ数では D の値は 2 を越えることはほとんどない。

### 5.3. 開発 LSI 仕様

上記アーキテクチャに基づき、今回 Longest Prefix Match 検索用チップ  $\mu$ PD98421 を開発した (写真 1 参照)。この LSI は LPM 検索の他に、従来の CAM と同様の完全一致 (Full Match) 検索と、マスク付き完全一致 (Full Match with Mask) 検索が可能である。表 3 に緒元を示す。

表 3:  $\mu$ PD98421 の緒元

	Full Match	Full Match with Mask	Longest Prefix Match
動作クロック	33MHz		
ビット幅	64bit		
エントリ数	8K	4K	4K
検索時間	1clock (30ns)	1 clock (30ns)	2 clocks (*) (60ns)
検索スループット (検索/秒)	33M	33M	16.5M(**)

\*4 チップ後続までの場合。

\*\*パイプライン検索をサポートしなかったため

本 LSI は検索能力が非常に高いだけでなく、エントリ更新は極めて簡単で、該当エントリを削除、あるいは追加するためには、LSI にプレフィクスとプレフィクス長 (マスクビット列) を格納するために 2 回書き込みを行うだけでよい。LPM 検索の場合、検索時間は、LSI の接続数によって異なり、以下ようになる。

#### 4 チップの場合 (16K エントリ)

- Wired OR delay D: 1clock
- 検索時間: 2 clocks (60ns)

#### 32 チップの場合 (128K エントリ)

- Wired OR delay D: 2clock
- 検索時間: 3 clocks (90ns)

現在のインターネットバックボーンルータのエントリ数は 50~60K と言われており、128K エントリはこれを十分収容できる。16K エントリ、128K エントリ、いずれの場合にも 4 章で示した検索時間の性能条件である 150ns を十分にクリアしており、複数の OC-48 リンクをワイヤースピード処理することが可能である。ただし、今回の開発ではパイプライン検索を省略した為、33MHz 動作で検索スループットは 16.5M 検索/秒となっている。しかし、OC-48 の 1 パケット時間に 2 回 LPM 検索をすることが可能である。

この LSI と 2.1 で述べたパイプライン処理によって、マルチギガビット IP フォワーダを実現する事が可能になる。

### 6. まとめ

本稿では、マルチギガビットフォワーダのためのパ

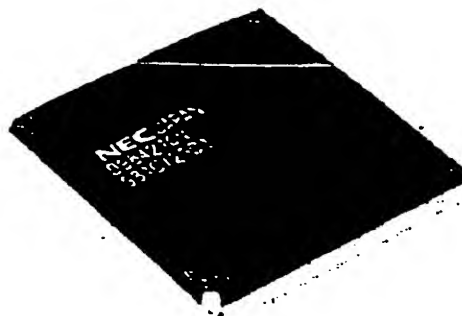


写真 1:  $\mu$ PD98421

イプライン化フォワーディング処理について述べ、必要とされる検索エンジンのアーキテクチャと実際に開発した検索 LSI の性能を示した。このアーキテクチャにより、最大 50Mpps の LPM 検索が可能となる。今回開発した LSI は 60ns で LPM 検索が可能で、最大 16.5M 検索/秒の検索能力を持つ。これを用いずば、最小パケット長を短く 48bytes と見積もっても、63Gbps のリンクにワイヤースピードで対応することが可能である。現在、パイプライン化検索など、さらなる高速化と大容量化を目指した製品を開発中である。

### 謝辞

最後に本研究に関して有益な助言を頂いた長谷川聡部長、谷英明主任研究員、立石久男マネージャに感謝致します。

### 参考文献

- [1] 豊島 他, "高速コンピュータシステムにおけるハードウェアベース高速フォワーディング方式の提案", 信学技報, SSE97-91(IN97-84, CS97-82), Sep. 1997.
- [2] 下西 他, "ATM 技術と IP 処理技術を融合した品質保証型スイッチアーキテクチャの提案", 1998 年電子情報通信学会ソサイエティ大会, SB-7-3, Sep. 1998.
- [3] Mikael Degermark, et al. "Small Forwarding Table for Fast Routing Lookups", in Proc. ACM SIGCOMM'97, Cannes, France, Sept. 1997.
- [4] Marcel Waldvoget, et al. "Scalable High Speed IP Routing Lookups", in Proc. ACM SIGCOMM'97, Cannes, France, Sept. 1997.
- [5] 林 他, "通信処理用高速テーブル検索モジュールの検討", 1998 年電子情報通信学会総合大会, B-6-121, 1998.
- [6] Kawasaki LSI, U.S.A., Inc., "Longest Match Engine KESBLME008", <http://www.lcsi.com/products/lme.html>
- [7] 小林, 村瀬 他, "高速マルチレイヤ処理 1 チップスイッチアーキテクチャ", 信学技報, SSE97-187(IN97-186), Mar. 1998.
- [8] Gray R. Wright, et al., TCP/IP Illustrated, Volume 2 The implementation, Addison-Wesley, 1995.